## AMENDMENTS TO THE CLAIMS

The claims have been amended as follows:

1. (Currently Amended) A computer-readable medium having stored thereon computer executable instructions to operate on a data structure identifying parameter value combinations, the instructions when executed causes a computer system ~~for use~~ to test a software module, the data structure comprising:

(a) a first section that includes a set of ~~testing~~ parameters listed in a parameter order for testing the software module;

(b) a second section, when instructed, ~~that includes~~ extracts a first set of parameter values and lists the first set of parameter values ~~listed~~ in an order such that each value in said first set of parameter values is positioned in the same order as the corresponding parameter ~~is~~ listed in the parameter order, wherein the first set of parameter values is identified with a first test case for testing the software module; and

(c) a third section, when instructed, ~~that includes~~ extracts a second set of parameter values and lists the second set of parameter values in ~~listed~~ an order such that each value in said second set of parameter values is positioned in the same order as the corresponding parameter ~~is~~ listed in the parameter order, wherein the second set of parameter values is identified with a second test case for testing the software module, .

2. (Original) The computer-readable medium of claim 1, wherein the testing parameters are marked up with a markup language.

3. (Original) The computer-readable medium of claim 2, wherein the markup language comprises the extensible markup language.

4. (Original) The computer-readable medium of claim 1, wherein the first section, second section and third section are elements of a table.

5. (Original) The computer-readable medium of claim 4, wherein the table comprises additional sections that include sets of parameter values.

6. (Currently Amended) A method of processing testing data for testing a software module, the method comprising:

(a) extracting parameter value combinations from a data file formatted with ~~marked up with~~ a markup language to implement data of an external table associated with a first test case;

(b) transmitting the parameter value combinations to a software module test engine, wherein the parameter value combinations are identified with the first test case; ~~and~~

(c) testing ~~a~~ the software module with the parameter value combinations based on the first test case;

(d) generating a first test result based on the first test case;

(e) changing the data file to implement data of the external table associated with a second test case for testing the software module, wherein the parameter value combinations are identified with the second test case; and

(f) generating a second test result based on the second test case.

7. (Currently Amended) The method of claim 6, wherein the ~~data file comprises a~~ external table ~~containing~~ comprises a plurality of test cases and each test case comprises a set of parameter value combinations.

8. (Original) The method of claim 7, wherein (a) comprises extracting the plurality of test cases from the data file.

9. (Original) The method of claim 7, further including creating an object from a test case parameter value combination.

10. (Original) The method of claim 6, further including changing the format of the parameter value combinations before (b).

11. (Original) The method of claim 6, further including:

(i) receiving a table of parameter value combinations at a spreadsheet application; and

(ii) converting the table to the data file with a spreadsheet plug-in.

12. (Original) The method of claim 6, further including validating the parameter value combinations by comparing the parameter value combinations to a set of rules.

13. (Original) The method of claim 12, wherein parameter value combinations are validated on demand prior to (b).

14. (Original) A computer-readable medium having computer executable instructions for performing the steps recited in claim 6.

15. (Original) A computer-readable medium having computer executable instructions for performing the steps recited in claim 11.

16. (Original) A computer-readable medium having computer executable instructions for performing the steps recited in claim 12.

17. (Currently Amended) A computer-readable medium containing computer-executable components comprising:

an import component that extracts parameter value combinations from a data file formatted with a marked up ~~with a markup~~ language to implement data of an external table associated with a first test case;

a test object creation module that creates an object to test a software module with the parameter value combinations associated with the first test case; wherein

the import component is configured to extract parameter value combinations from the data file to implement data of the external table associated with a second test case for testing the software module.

18. (Original) The computer-readable medium of claim 17, wherein the markup language comprises the extensible markup language.

19. (Original) The computer-readable medium of claim 17, wherein the import module validates the parameter value combinations.

20. (Cancelled)

21. (Cancelled)

22. (New) A computer-readable medium having stored thereon computer executable program for testing a software module, the computer program when executed causes a computer system to execute steps of:

(a) extracting a set of parameters listed in a parameter order for testing the software module;

(b) extracting a first set of parameter values and listing the first set of parameter values in an order such that each value in said first set of parameter values is positioned in the same order as the corresponding parameter listed in the parameter order, wherein the first set of parameter values is identified with a first test case for testing the software module;

(c) testing the software module based on the first test case;

(d) extracting a second set of parameter values and listing the second set of parameter values in an order such that each value in said second set of parameter values

is positioned in the same order as the corresponding parameter listed in the parameter order, wherein the second set of parameter values is identified with a second test case for testing the software module; and

(e) testing the software module based on the second test case.